

Unifying Registration based Tracking: A Case Study with Structural Similarity

Abhineet Singh

Mennatullah Siam
University of Alberta

Martin Jagersand

asinghl, mennatul, mj7@ualberta.ca

Abstract

This paper adapts a popular image quality measure called structural similarity for high precision registration based tracking while also introducing a simpler and faster variant of the same. Further, these are evaluated comprehensively against existing measures using a unified approach to study registration based trackers that decomposes them into three constituent sub modules - appearance model, state space model and search method. Several popular trackers in literature are broken down using this method so that their contributions - as of this paper - are shown to be limited to only one or two of these submodules.

An open source tracking framework is made available that follows this decomposition closely through extensive use of generic programming. It is used to perform all experiments on four publicly available datasets so the results are easily reproducible. This framework provides a convenient interface to plug in a new method for any sub module and combine it with existing methods for the other two. It can also serve as a fast and flexible solution for practical tracking needs due to its highly efficient implementation.

1. Introduction

Visual tracking is an important field in computer vision with diverse application domains including robotics, surveillance, targeting systems, autonomous navigation and augmented reality. Registration based tracking (**RBT**), also known in literature as direct visual tracking [59, 53, 49, 47], is a sub field thereof where the object is tracked by warping each image in a sequence to align the object patch with the template. Trackers of this type are especially popular in robotics and augmented reality applications both because they estimate the object pose with greater precision and are significantly faster than online learning and detection based trackers (**OLTs**) [62].

However, OLTs are better suited to general purpose tracking as RBTs are prone to failure when the object undergoes significant appearance changes due to factors like occlusions and lighting variations or when completely novel

views of the object are presented by deformations or large pose changes. As a result, OLTs are more popular in literature and have been the subject of several recent studies [40, 63, 36]. The scope of such studies is usually limited to finding the trackers that work best under challenging conditions by testing them on representative sequences with little to no analysis provided regarding *why* specific trackers work better than others for given challenges. This is understandable since such trackers differ widely in design and have little in common that may be used to relate them to each other and perform comparative analysis from a design perspective.

As we show in this work, however, RBTs do not have this drawback and can be decomposed into three sub modules - appearance model (**AM**), state space model (**SSM**) and search method (**SM**) (Sec. 2.2) - that makes their systematic analysis feasible. Though this decomposition is somewhat obvious and indeed has been observed before [64, 49], it has never been explored systematically or used to improve the study of this paradigm of tracking. It is the intent of this work to fill this gap by unifying the myriad of contributions made in this domain since the original Lucas Kanade (LK) tracker was introduced [44].

Most of these works have presented novel ideas for only one of these submodules while using existing methods, often selected arbitrarily, for the rest. For instance, Hager & Belhumeur [25], Shum & Szeliski [55], Baker & Matthews [3] and Benhimane & Malis [8] all introduced new variants of the Newton type SM used in [44] but only tested these with SSD¹ AM. Similarly, Dick et. al [19] and Zhang et. al [71] only combined their stochastic SMs with RSCV and SSD respectively. Conversely, Richa et. al [48], Scandaroli et. al [53] and Dame et. al [17] introduced SCV, MI and NCC as AMs but combined these only with a single SM - ESM in the former and ICLK in the other two. Even more recent works that use illumination models (**ILM**) (Sec. 2.2.1), including Bartoli [5, 6], Silvera & Malis [57, 58, 60] and Silvera [56], have combined their respective ILMs with only a single SM in each case. Finally, most SMs and AMs have been tested with only one SSM -

¹refer Sec. 2.2 for acronyms

either homography [8, 9, 17] or affine [3, 50]. In fact, Benhimane & Malis [8] mentioned that their SM only works with $\mathbb{SL}(3)$ SSM though experiments (Sec. 4.4) showed that it works equally well with others.

Such limited testing might not only give false indications about a method’s capability but might also prevent it from achieving its full potential. For instance, an AM that outperforms others with a given SM might not do so with other SMs and an SM may perform better with an AM other than the one it is tested with. In such cases, our decomposition can be used to experimentally find the optimal combination of methods for any contribution while also providing novel insights about its workings. We demonstrate its practical applicability by comparing several existing methods for these sub modules not only with each other but also with two new AMs based on structural similarity (SSIM) [69] that we introduce here and fit within this framework.

SSIM is a popular measure for evaluating the quality of image compression algorithms by comparing the compressed image with the original one. Since it measures the information loss in the former - essentially a slightly distorted or damaged version of the latter - it makes a suitable metric for comparing candidate warped patches with the template to find the one with the minimum loss and thus most likely to represent the same object. Further, it has been designed to capture the perceptual similarity of images and is known to be robust to illumination and contrast changes [69]. It is reasonable, therefore, to expect it to perform well for tracking under challenging conditions too. As such, it has indeed been used for tracking before with particle filters [41, 68], gradient ascent [43] and hybrid [42] SMs. All of these trackers, however, used imprecise SSMs with low degrees of freedom (DOF) - estimating only translation and scaling of the target patch. To the best of our knowledge, no attempt has been made to use SSIM for high DOF RBT within the LK framework [44]. This work aims to fill this gap too.

To summarize, following are the main contributions of this work:

- Adapt a popular image quality measure - SSIM - for high precision RBT and introduce a simpler but faster version called SPSS (Sec. 3).
- Evaluate these models comprehensively by comparing against 8 existing AMs using 11 SMs and 7 SSMs. Experiments are done using 4 large datasets with over 100,000 frames in all to ensure their statistical significance.
- Compare low DOF RBTs against state of the art OLTs to validate the suitability of the former for fast and high precision tracking applications.

- Provide an open source tracking framework² called MTF [61] using which all results can be reproduced and which, owing to its efficient C++ implementation, can also serve as a practical tracking solution.

Rest of this paper is organized as follows - section 2 describes the decomposition, section 3 presents details about SSIM and section 4 provides results and analysis.

2. Background

2.1. Notation

Let $I_t : \mathbb{R}^2 \mapsto \mathbb{R}$ refer to an image captured at time t treated as a function of real values using sub pixel interpolation [16] for non integral locations. The patch corresponding to the tracked object’s location in I_t is denoted by $\mathbf{I}_t(\mathbf{x}_t) \in \mathbb{R}^N$ where $\mathbf{x}_t = [\mathbf{x}_{1t}, \dots, \mathbf{x}_{Nt}]$ with $\mathbf{x}_{kt} = [x_{kt}, y_{kt}]^T \in \mathbb{R}^2$ being the Cartesian coordinates of pixel k . Further, $\mathbf{w}(\mathbf{x}, \mathbf{p}_s) : \mathbb{R}^2 \times \mathbb{R}^S \mapsto \mathbb{R}^2$ denotes a warping function of S parameters that represents the set of allowable image motions of the tracked object by specifying the deformations that can be applied to \mathbf{x}_0 to align $\mathbf{I}_t(\mathbf{w}(\mathbf{x}_0, \mathbf{p}_{st}))$ with $\mathbf{I}_0(\mathbf{x}_0)$. Finally $f(\mathbf{I}^*, \mathbf{I}^c, \mathbf{p}_a) : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^A \mapsto \mathbb{R}$ is a function of A parameters that measures the similarity between two patches - the reference or template patch \mathbf{I}^* and a candidate patch \mathbf{I}^c . For brevity, $\mathbf{I}_t(\mathbf{w}(\mathbf{x}_0, \mathbf{p}_{st}))$ and $\mathbf{I}_0(\mathbf{x}_0)$ will also be denoted as \mathbf{I}_t and \mathbf{I}_0 respectively.

2.2. Decomposing registration based tracking

Using this notation, RBT can be formulated as a search problem whose aim is to find the optimal parameters $\mathbf{p}_t = [\mathbf{p}_{st}, \mathbf{p}_{at}] \in \mathbb{R}^{S+A}$ that maximize the similarity, given by f , between $\mathbf{I}^* = \mathbf{I}_0$ and $\mathbf{I}^c = \mathbf{I}_t$, that is,

$$\mathbf{p}_t = \underset{\mathbf{p}_s, \mathbf{p}_a}{\operatorname{argmax}} f(\mathbf{I}_0(\mathbf{x}_0), \mathbf{I}_t(\mathbf{w}(\mathbf{x}_0, \mathbf{p}_s)), \mathbf{p}_a) \quad (1)$$

This formulation gives rise to an intuitive way to decompose the tracking task into three sub modules - the similarity metric f , the warping function \mathbf{w} and the optimization approach. These can be designed to be semi independent in the sense that any given optimizer can be applied unchanged to several combinations of methods for the other two modules which in turn interact only through a well defined and consistent interface. In this work, these sub modules are respectively referred to as AM, SSM and SM. A more detailed description with examples follows.

2.2.1 Appearance Model

This is the image similarity metric defined by f in Eq. 1 that the SM uses to compare different warped patches from I_t to find the closest match with \mathbf{I}^* . It may be noted that

²available at <http://webdocs.cs.ualberta.ca/~vis/mtf/>

\mathbf{I}^* is not constrained to be fixed but may be updated during tracking as in [50]. Examples of f with $A = 0$ include sum of squared differences (SSD) [44, 4, 9], sum of conditional variance (SCV) [48], reversed SCV (RSCV) [19], normalized cross correlation (NCC) [53], mutual information (MI) [20, 17] and cross cumulative residual entropy (CCRE) [67, 49]. There has also been a recent extension to SCV called LSCV [47] that claims to handle localized intensity changes better. Finally, it has been shown [52] that SSD performs better when applied to z-score [32] normalized images which makes it equivalent to NCC. The resultant formulation is considered here as a distinct AM called Zero Mean NCC (ZNCC). These AMs can be divided into 2 categories - those that use some form of the L2 norm as f - SSD, SCV, RSCV, LSCV and ZNCC - and those that do not - MI, CCRE and NCC. The latter are henceforth called robust AMs after [49]. The two AMs introduced here - SSIM and SPSS - too fall into this category.

To the best of our knowledge, the only AMs with $A \neq 0$ introduced thus far in literature are those with an illumination model (ILM), where f is expressed as $f(\mathbf{I}^*, g(\mathbf{I}^c, \mathbf{p}_a))$ with $g : \mathbb{R}^N \times \mathbb{R}^A \mapsto \mathbb{R}^N$ accounting for differences in lighting conditions under which I_0 and I_t were captured. These include gain & bias (GB) [5, 6] and piecewise gain & bias (PGB) [57, 58, 60, 56] with the latter comprising surface modeling with radial basis function (RBF) too. Though there have also been AMs that incorporate online learning [33, 50, 21], they are not compatible with all SMs used here and so have not been tested.

2.2.2 State Space Model

This is the warping function \mathbf{w} that represents the deformations that the tracked patch can undergo. It therefore defines the set of allowable image motions of the corresponding object and can be used to place constraints on the search space of \mathbf{p}_s to make the optimization more robust or efficient. Besides the DOF of allowed motion, SSM also includes the actual parameterization of \mathbf{w} . For instance, even though both represent 8 DOF motion, $\mathbb{SL}(3)$ [9, 37] is considered as a different SSM than standard homography [44, 4] that uses actual entries of the corresponding matrix.

This work uses 7 SSMs including 5 from the standard hierarchy of geometrical transformations [27, 64] - translation, isometry, similitude, affine and homography - along with two alternative parameterizations of homography - $\mathbb{SL}(3)$ and corner based (i.e. using x,y coordinates of the bounding box corners). More complex SSMs to handle non rigid objects have also been proposed in literature like thin plate splines [10], basis splines [65] and quadric surfaces [54]. However, these are not tested here as the datasets used only feature rigid objects so these cannot be fairly evaluated. Extensions like incorporation of 3D pose [14] and

camera parameters [66] to \mathbf{w} are also excluded.

2.2.3 Search Method

This is the optimization procedure that searches for the warped patch in I_t that best matches \mathbf{I}^* . There have been two main categories of SMs in literature - gradient descent (GD) and stochastic search. The former category includes the four variants of the classic Lucas Kanade (LK) tracker [44] - forward additive (FALK) [44], inverse additive (IALK) [25], forward compositional (FCLK) [64] and inverse compositional (ICLK) [3] - that have been shown [3, 4] to be equivalent to first order terms. Here, however, we show experimental results (Sec. 4.4.3) proving that they perform differently in practice. A more recent approach of this type is the Efficient Second order Minimization (ESM) [9] technique that uses gradients from both I_0 and I_t to make the best of ICLK and FCLK. Several extensions have also been proposed to these SMs to handle specific challenges like motion blur [46], resolution degradation [31], better Taylor series approximation [35] and optimal subset selection [7]. Though these can be fit within our framework, either as distinct SMs or as variants thereof, they are not considered here for lack of space.

There are three main stochastic SMs in literature - approximate nearest neighbor (NN) [24, 19], particle filters (PF) [30, 38, 39, 21, 37] and random sample consensus (RANSAC) [12, 71]. Though less prone to getting stuck in local maxima than GD, their performance depends largely on the number and quality of random samples used and tends to be rather jittery and unstable due to the limited coverage of the search space. One approach to obtain better precision is to combine them with GD methods [19, 71] where results from the former are used as starting points for the latter. Three such combinations have been tested here as examples of hybrid or composite SMs - NN+ICLK (NNIC), PF+FCLK (PFFC) and RANSAC+FCLK (RKLT).

3. Structural Similarity

SSIM was originally introduced [69] to assess the loss in image quality incurred by compression methods like JPEG. It has been very popular in this domain since it closely mirrors the approach adopted by the human visual system to subjectively evaluate the quality of an image. SSIM between \mathbf{I}_0 and \mathbf{I}_t is defined as a product of 3 components:

$$f_{ssim} = \left(\frac{2\mu_t\mu_0 + C_1}{\mu_t^2 + \mu_0^2 + C_1} \right)^\alpha \left(\frac{2\sigma_t\sigma_0 + C_2}{\sigma_t^2 + \sigma_0^2 + C_2} \right)^\beta \left(\frac{\sigma_{t0} + C_3}{\sigma_t\sigma_0 + C_3} \right)^\gamma \quad (2)$$

where μ_t is the mean and σ_t is the sample standard deviation of \mathbf{I}_t while σ_{t0} is the sample covariance between \mathbf{I}_t and \mathbf{I}_0 . The three components of f_{ssim} from left to right are respectively used for luminance, contrast and structure com-

parison between the two patches. The positive constants α, β, γ are used to assign relative weights to these components while C_1, C_2, C_3 are added to ensure their numerical stability with small denominators. Here, as in most practical implementations [69, 41, 42, 43, 68], it is assumed that $\alpha = \beta = \gamma = 1$ and $C_3 = \frac{C_2}{2}$ so that Eq. 2 simplifies to:

$$f_{ssim} = \frac{(2\mu_t\mu_0 + C_1)(2\sigma_{t0} + C_2)}{(\mu_t^2 + \mu_0^2 + C_1)(\sigma_t^2 + \sigma_0^2 + C_2)} \quad (3)$$

3.1. Newton's Method with SSIM

Using SSIM with NN and PF is straightforward since these only need f_{ssim} to be computed between candidate patches. However, GD SMs also require its derivatives as they solve Eq 1 by estimating an incremental update $\Delta \mathbf{p}_t$ to \mathbf{p}_{t-1} using some variant of Newton's method as:

$$\Delta \mathbf{p}_t = -\hat{\mathbf{H}}^{-1} \hat{\mathbf{J}}^T \quad (4)$$

where $\hat{\mathbf{J}}$ and $\hat{\mathbf{H}}$ respectively are estimates for Jacobian $\mathbf{J} = \partial f / \partial \mathbf{p}$ and Hessian $\mathbf{H} = \partial^2 f / \partial \mathbf{p}^2$ of f w.r.t. \mathbf{p} . These can be further decomposed using chain rule as:

$$\mathbf{J} = \frac{\partial f}{\partial \mathbf{I}} \frac{\partial \mathbf{I}}{\partial \mathbf{p}} = \frac{\partial f}{\partial \mathbf{I}} \nabla \mathbf{I} \frac{\partial \mathbf{w}}{\partial \mathbf{p}} \quad (5)$$

$$\mathbf{H} = \frac{\partial \mathbf{I}^T}{\partial \mathbf{p}} \frac{\partial^2 f}{\partial \mathbf{I}^2} \frac{\partial \mathbf{I}}{\partial \mathbf{p}} + \frac{\partial f}{\partial \mathbf{I}} \frac{\partial^2 \mathbf{I}}{\partial \mathbf{p}^2} \quad (6)$$

Of the right hand side terms in Eqs. 5 and 6, only $\partial f / \partial \mathbf{I}$ and $\partial^2 f / \partial \mathbf{I}^2$ depend on f so the relevant expressions for f_{ssim} are presented below - general formulations corresponding to \mathbf{J} and \mathbf{H} in Eqs. 7 and 8 respectively, followed by specializations for $\hat{\mathbf{J}}$ and $\hat{\mathbf{H}}$ used by specific SMs. Detailed derivations are presented in the supplementary.

$$\frac{\partial f_{ssim}}{\partial \mathbf{I}_t} = \mathbf{f}' = \frac{2}{cd} \left[\left(\frac{a\bar{\mathbf{I}}_0 - cf\bar{\mathbf{I}}_t}{N-1} + \frac{\mu_0 b - \mu_t f d}{N} \right) \right] \quad (7)$$

$$\begin{aligned} \frac{\partial^2 f_{ssim}}{\partial \mathbf{I}_t^2} = & \frac{2}{cd} \left[\frac{1}{N} S_N \left(\frac{4}{N-1} (\mu_0 \bar{\mathbf{I}}_0 - \mu_t f \bar{\mathbf{I}}_t) - \frac{3\mu_t d}{2} \mathbf{f}' - \frac{f d}{N} \right) \right. \\ & \left. - \frac{c}{N-1} \left(\frac{3}{2} \mathbf{f}'^T \bar{\mathbf{I}}_t + f \mathbb{I} \right) \right] \quad (8) \end{aligned}$$

with $\bar{\mathbf{I}}_t = \mathbf{I}_t - \mu_t$, $a = 2\mu_t\mu_0 + C_1$, $b = 2\sigma_{t0} + C_2$, $c = \mu_t^2 + \mu_0^2 + C_1$, $d = \sigma_t^2 + \sigma_0^2 + C_2$, $f = f_{ssim}$ and $S_n(\mathbf{K})$ denoting an $n \times k$ matrix formed by stacking the $1 \times k$ vector \mathbf{K} into rows. The form of $\partial f / \partial \mathbf{I}$ used by the four variants of LK is identical except that ICLK requires the differentiation to be done w.r.t. \mathbf{I}_0 instead of \mathbf{I}_t - the expressions for this are trivial to derive since SSIM is symmetrical. ESM was originally [9] formulated as using the mean of $\nabla \mathbf{I}_0$ and $\nabla \mathbf{I}_t$ to compute \mathbf{J} but, as this formulation is only applicable to SSD, a generalized version [11, 53] is considered here that uses the *difference* between FCLK and ICLK Jacobians.

It is generally assumed [4, 9] that the second term of Eq. 6 is too costly to compute and too small near convergence to matter and so is omitted to give the Gauss Newton Hessian:

$$\mathbf{H}_{gn} = \frac{\partial \mathbf{I}^T}{\partial \mathbf{p}} \frac{\partial^2 f}{\partial \mathbf{I}^2} \frac{\partial \mathbf{I}}{\partial \mathbf{p}} \quad (9)$$

Though \mathbf{H}_{gn} works very well for SSD (and in fact even better than \mathbf{H} [4, 16]), it is well known [16, 53] to *not* perform well with other AMs like MI, CCRE and NCC and we can confirm that the latter is true for SSIM and SPSS too. For these AMs, an approximation to \mathbf{H} *after convergence* has to be used instead by assuming perfect alignment between the patches or $\mathbf{I}^c = \mathbf{I}^*$. This approximation is here referred to as the **Self Hessian** and, as this substitution can be made by setting either $\mathbf{I}^c = \mathbf{I}_0$ or $\mathbf{I}^* = \mathbf{I}_t$, we get two forms which are respectively deemed to be ICLK and FCLK Hessians:

$$\hat{\mathbf{H}}_{ic} = \mathbf{H}_{self}^* = \frac{\partial \mathbf{I}_0^T}{\partial \mathbf{p}} \frac{\partial^2 f(\mathbf{I}_0, \mathbf{I}_0)}{\partial \mathbf{I}^2} \frac{\partial \mathbf{I}_0}{\partial \mathbf{p}} + \frac{\partial f(\mathbf{I}_0, \mathbf{I}_0)}{\partial \mathbf{I}} \frac{\partial^2 \mathbf{I}_0}{\partial \mathbf{p}^2} \quad (10)$$

$$\hat{\mathbf{H}}_{fc} = \mathbf{H}_{self}^c = \frac{\partial \mathbf{I}_t^T}{\partial \mathbf{p}} \frac{\partial^2 f(\mathbf{I}_t, \mathbf{I}_t)}{\partial \mathbf{I}^2} \frac{\partial \mathbf{I}_t}{\partial \mathbf{p}} + \frac{\partial f(\mathbf{I}_t, \mathbf{I}_t)}{\partial \mathbf{I}} \frac{\partial^2 \mathbf{I}_t}{\partial \mathbf{p}^2} \quad (11)$$

It is interesting to note that \mathbf{H}_{gn} has the exact same form as \mathbf{H}_{self} for SSD (since $\partial f_{ssd}(\mathbf{I}_0, \mathbf{I}_0) / \partial \mathbf{I} = \partial f_{ssd}(\mathbf{I}_t, \mathbf{I}_t) / \partial \mathbf{I} = 0$) so it seems that interpreting Eq. 9 as the first order approximation of Eq. 6 for SSD as in [4, 20, 16] is incorrect and it should instead be seen as a special case of \mathbf{H}_{self} . Setting $\mathbf{I}_0 = \mathbf{I}_t$ simplifies Eq. 8 to:

$$\frac{\partial^2 f_{ssim}(\mathbf{I}_t, \mathbf{I}_t)}{\partial \mathbf{I}_t^2} = \frac{-2}{\bar{c}\bar{d}} \left[\frac{\bar{d}}{N^2} + \frac{\bar{c}}{N-1} \mathbb{I} \right] \quad (12)$$

with $\bar{c} = 2\mu_t^2 + C_1$ and $\bar{d} = 2\sigma_t^2 + C_2$. Finally, FALK and IALK use the same form of $\partial^2 f / \partial \mathbf{I}^2$ as FCLK while ESM uses the *sum* of $\hat{\mathbf{H}}_{fc}$ and $\hat{\mathbf{H}}_{ic}$.

3.2. Simplifying SSIM with pixelwise operations

In the formulation described so far, SSIM has been computed over the *entire* patch - i.e. μ_t, σ_t and σ_{t0} have been computed over all N pixels in \mathbf{I}_t and \mathbf{I}_0 . In its original form [69], however, the expression in Eq. 3 was applied to several corresponding *sub windows* within the two patches - for instance 8×8 sub windows that are moved pixel-by-pixel over the entire patch - and the *mean* of all resultant scores taken as the overall similarity score. For tracking applications, such an approach is not only impracticable from speed perspective, it presents another issue for GD SMs - presence of insufficient texture in these small sub windows may cause Eq. 4 to become ill posed if \mathbf{J} and \mathbf{H} are computed for each sub window and then averaged.

As a result, the formulation used here considers only one end of the spectrum of variation of size and number of sub windows - a single sub window of the same size as the patch. Now, if the *other* end of the spectrum is considered - N sub windows of size 1×1 each - then a different

AM is obtained that may provide some idea about the effect of window wise operations while also being much simpler and faster. The resultant model is termed as **Sum of Pixelwise Structural Similarity** or **SPSS**. When considered pixel wise, σ_t and σ_{t0} become null while μ_t becomes equal to the pixel value itself so that Eq. 3 simplifies to:

$$f_{spss} = \sum_{i=1}^N \frac{2\mathbf{I}_t(\mathbf{x}_{it})\mathbf{I}_0(\mathbf{x}_{i0}) + C_1}{\mathbf{I}_t(\mathbf{x}_{it})^2 + \mathbf{I}_0(\mathbf{x}_{i0})^2 + C_1} \quad (13)$$

Similar to SSD, contributions from different pixels to f_{spss} are independent of each other so that each entry of $\partial f_{spss} / \partial \mathbf{I}_t$ has contribution only from the corresponding pixel. This also holds true for each entry of the principal diagonal of $\partial^2 f_{spss} / \partial \mathbf{I}_t^2$ (which is a diagonal matrix). Denoting the contributions of the i^{th} pixel to f_{spss} , $\partial f_{spss} / \partial \mathbf{I}_t$ and $\partial^2 f_{spss} / \partial \mathbf{I}_t^2$ respectively as f_i , f'_i and f''_i , we get:

$$f'_i = \frac{2(\mathbf{I}_0(\mathbf{x}_{i0}) - \mathbf{I}_t(\mathbf{x}_{it})f_i)}{\mathbf{I}_t(\mathbf{x}_{it})^2 + \mathbf{I}_0(\mathbf{x}_{i0})^2 + C_1} \quad (14)$$

$$f''_i = \frac{-2(f_i + 3\mathbf{I}_t(\mathbf{x}_{it})f'_i)}{\mathbf{I}_t(\mathbf{x}_{it})^2 + \mathbf{I}_0(\mathbf{x}_{i0})^2 + C_1} \quad (15)$$

$$f''_i(\mathbf{I}_t, \mathbf{I}_t) = \frac{-2}{2\mathbf{I}_t(\mathbf{x}_{it})^2 + C_1} \quad (16)$$

4. Results and Analysis

4.1. Datasets

Following four publicly available datasets have been used to analyze the trackers:

1. **Tracking for Manipulation Tasks (TMT)** dataset [51] that contains videos of some common tasks performed at several speeds and under varying lighting conditions. It has 109 sequences with 70592 frames.
2. **Visual Tracking Dataset** provided by **UCSB** [23] that has 96 short sequences with a total of 6889 frames. These are more challenging than TMT but also somewhat artificial as they were created to represent specific challenges rather than realistic tasks.
3. **LinTrack** dataset [72] that has 3 long sequences with a total of 12477 frames. These are more realistic than those in UCSB but also more difficult to track.
4. A collection of 28 challenging planar tracking sequences from several significant works in literature [37, 57, 58, 60, 56, 15, 45]. We call this the **PAMI** dataset after [37] from where several of the sequences originate. There are 16511 frames in all.

All of these datasets except PAMI have full pose (8 DOF) ground truth data which makes them suitable for evaluating high precision trackers that are the subject of this study. For PAMI, this data was generated using a combination of very high precision tracking and manual refinement [51].

4.2. Evaluation Measures

Alignment Error (E_{AL}) [19] has been used as the metric to compare tracking result with the ground truth since it accounts for fine misalignments of pose better than other measures like center location error and Jaccard index. A tracker's overall accuracy is measured through its **success rate** (SR) which is defined as the fraction of total frames where E_{AL} is less than a threshold of t_p pixels. Formally, $SR = |S|/|F|$ where $S = \{f^i \in F : E_{AL}^i < t_p\}$, F is the set of all frames and E_{AL}^i is the error in the i^{th} frame f^i .

Since there are far too many sequences to present results for each, an overall summary of performance is reported instead by averaging the SR over all sequences in the four datasets. In addition, to better utilize frames that follow a tracker's first failure in any sequence, we initialize trackers at 10 different evenly spaced frames in each sequence. Therefore the SR plots represent accumulated tracking performance over a total of $|F| = 589380$ frames, out of which 106469 are unique.

Finally, the SR is evaluated for several values of t_p ranging from 0 to 20 and the resulting SR vs. t_p plot is studied to get an overall idea of how precise and robust a tracker is. As an alternative measure for robustness, reinitialization tests [36] were also conducted where a tracker is reinitialized after skipping 5 frames every time its E_{AL} exceeds 20 and the number of such reinitialization is counted. Due to space constraints, these results are presented in the supplementary (available on MTF website).

4.3. Parameters Used

All results were generated using a fixed sampling resolution of 50×50 irrespective the tracked object's size. Input images were smoothed using a Gaussian filter with a 5×5 kernel before being fed to the trackers. Iterative SMs were allowed to perform a maximum of 30 iterations per frame but only as long as the L2 norm of the change in bounding box corners in each iteration exceeded 0.0001. Implementation details of NN, PF and RANSAC were taken from [19], [37] and [71] respectively. NN was run with 1000 samples and PF with 500 particles. For RANSAC and RKLT, a 10×10 grid of sub patches was used and each sub patch was tracked by a 2 DOF FCLK tracker with a sampling resolution of 25×25 . As in [69], SSIM parameters are computed as $C_1 = (K_1 L)^2$ and $C_2 = (K_2 L)^2$ with $K_1 = 0.01$, $K_2 = 0.03$ and $L = 255$. Learning based trackers (Sec. 4.4.2) were run using default settings provided by their respective authors. Speed tests were performed on a 4 GHz Intel Core i7-4790K machine with 16 GB of RAM.

4.4. Results

The results presented in this section are organized into three sections corresponding to the three sub modules. In each, we present and analyze results comparing different

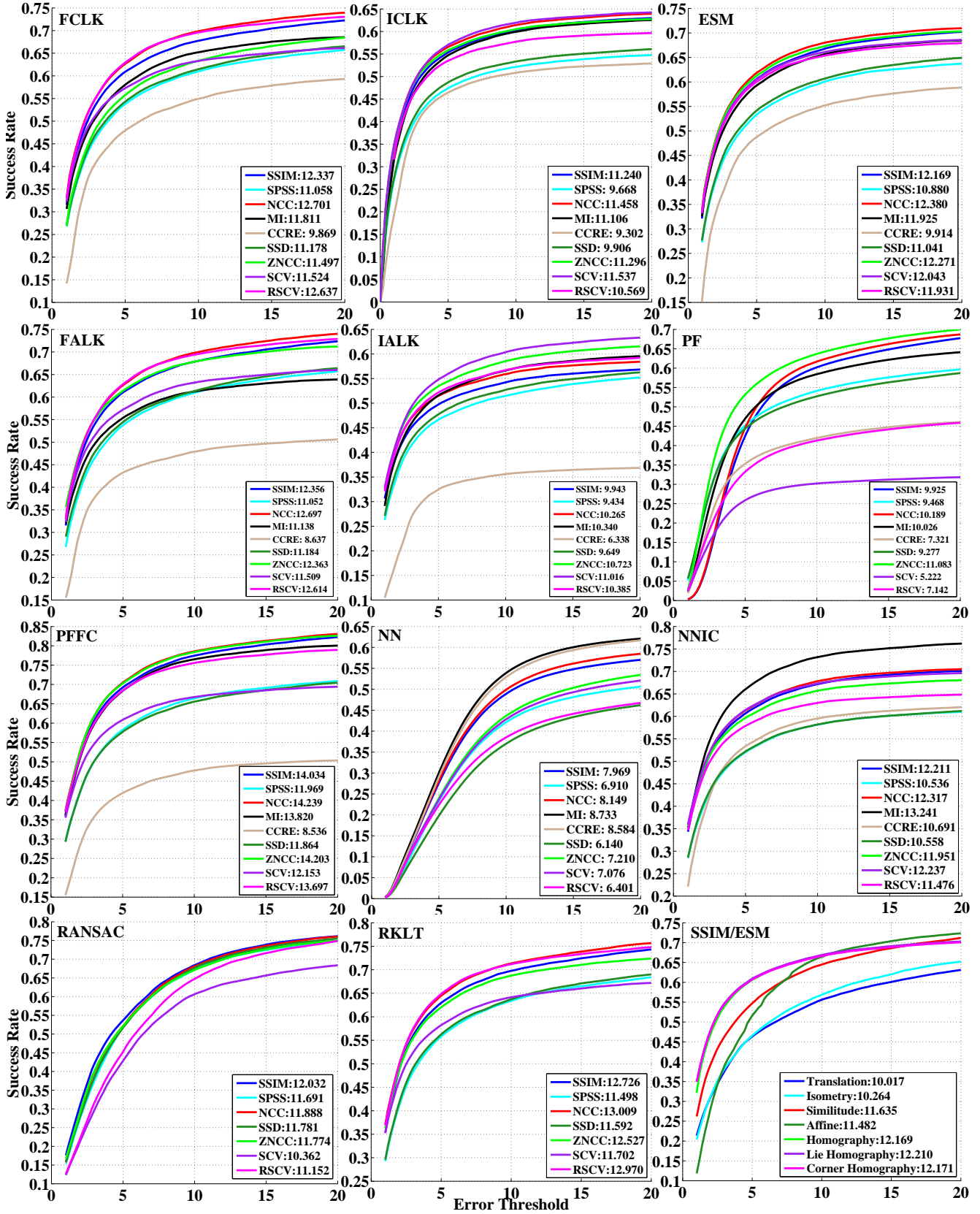


Figure 1: Success rates for AMs using different SMs with Homography as well as for different SSMs using SSIM with ESM (bottom right). Plot legends indicate areas under the respective curves. Best viewed on a high resolution screen.

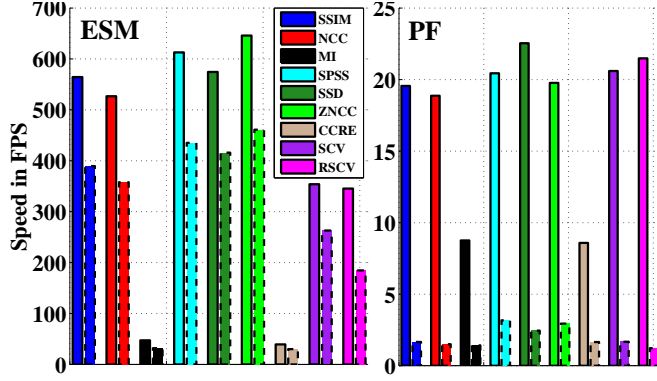


Figure 2: Speeds of ESM and PF with homography. Solid and dotted lines show the means and standard deviations respectively.

methods for the respective sub module.

4.4.1 Appearance Models

Fig. 1 shows the SR curves for all AMs using different SMs and homography SSM with plot legends also indicating the areas under the respective curves which are equivalent [13] to the average SR over this range of t_p . Fig. 2 shows the speeds of these AMs with ESM and PF as representatives of GD and stochastic SMs respectively. LSCV is excluded to reduce clutter as it was found to perform very similarly to SCV. Its results are in the supplementary instead along with those for the ILMs. Further, RANSAC and RKLT do not include CCRE and MI results since both are far too slow for 100 of them to be run simultaneously in real time. Also, both perform *very* poorly with these SMs, probably because the small sub patches used there [71] do not have sufficient information to be represented well by the joint histograms employed by these AMs.

Several observations can be made here. Firstly, NCC is the best performer with all SMs except IALK and NN - IALK performs poorly with all robust AMs (Sec. 4.4.3) while NN performs best with MI. Nevertheless, SSIM is usually equivalent to NCC or a close second showing that the latter is among the best AMs known. Also, as expected, SSIM is much better than SPSS with all SMs, with the latter only managing to match the performance of SSD on average. Further, though ZNCC is claimed to be equivalent to NCC [52] and also has a wider basin of convergence due to its SSD like formulation, it usually does *not* perform as well as NCC.

Secondly, in spite of being the most sophisticated and computationally expensive AM, CCRE is the worst performer with GD SMs and even MI is only slightly better than SSD on average. However, MI and CCRE are actually the best performing AMs with NN and MI is so with NNIC too. This shows that their poor performance with GD SMs

is likely to be due to their narrower basin of convergence rather than an inherent weakness in the AMs themselves. This also explains MI's significant lead over CCRE with these SMs though the two differ only in the latter using a cumulative joint histogram. It seems likely that the additional complexity of CCRE along with the resultant invariance to appearance changes further *reduces* its basin of convergence [17]. This discrepancy in performance between GD and stochastic SMs demonstrates the inadequacy of evaluating an AM with only one SM.

Thirdly, SCV outperforms RSCV with both inverse GD SMs - ICLK and IALK - though the reverse is true with both the forward ones - FCLK and FALK. This pattern also holds for stochastic and composite SMs - SCV is better with NN/NNIC where samples are generated from I_0 but worse with PF/PFFC where these come from I_t . Also, their performance is very similar with ESM where information from both I_0 and I_t is used. This is probably because SCV performs likelihood substitution with I_0 [48] while RSCV does so with I_t [19]. Fourthly, the separation between AMs is narrower with RANSAC than other SMs as this SM depends more on the *number* of sub patches used than on the tracker used for each. Conversely, PF shows maximum variation between AMs, thus indicating its strong reliance on f .

Finally, SPSS is not much faster than SSIM with either SM though it has lower computational complexity. This is partly due to SSIM finding convergence in fewer iterations with GD SMs and partly due to the way Eigen [1] optimizes matrix multiplications, many of which are used for computing f_{ssim} and its derivatives while those of f_{spss} have to be computed pixel by pixel. The same holds for SSD too.

4.4.2 State Space Models

Results in this section follow a slightly different format from the other two due to the difference in the motivations for using low DOF SSMs - they are more robust due to reduced search space and faster due to the gradients of w being less expensive to compute. Limiting the DOF also makes them directly comparable to OLTs which too have low DOF. As a result, 2 DOF RBTs were also tested with 8 state of the art OLTs [36] - DSST [18], KCF [29], CMT [45], RCT [70], TLD [34], Struck [26], FRG [2] and GO-TURN [28]. Lastly, in order to make the evaluations fair, *lower DOF ground truth* has been used for all accuracy results in this section. This was generated for each SSM using least squares optimization to find the warp parameters that, when applied to the initial bounding box, will produces a warped box whose E_{AL} with respect to the 8 DOF ground truth is as small as it is possible to achieve given the constraints of that SSM.

Fig. 3 shows the results of these tests in terms of both accuracy and speed. As expected, all the OLTs have low

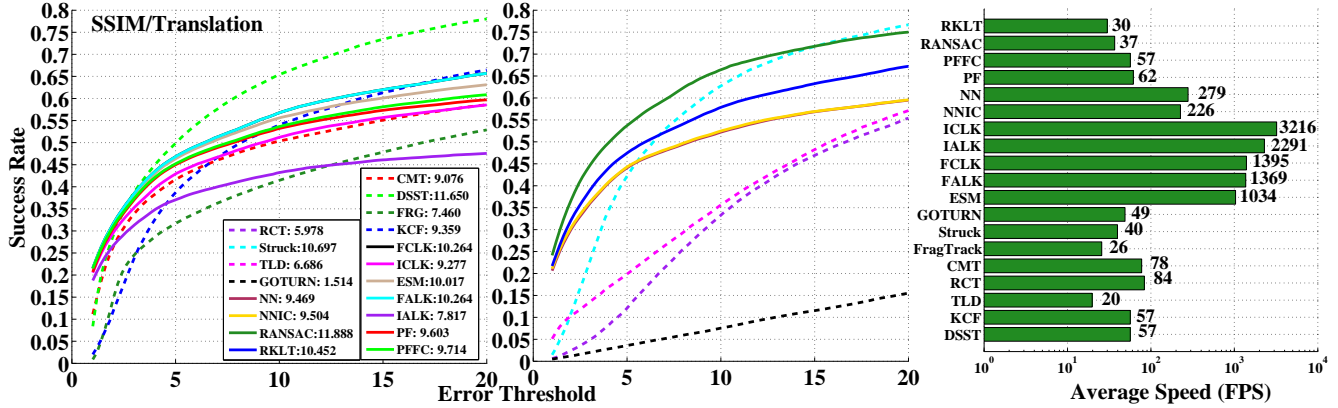


Figure 3: Success Rates for SSIM using translation as well as for 8 OLTs. The former are shown with **solid** lines and the latter in **dashed** lines. The speed plot on the right has **logarithmic scaling** on the x axis for clarity though actual figures are also shown.

SR for smaller t_p since they are less precise in general [36]. What is more interesting, however, is that none of them, with the exception of DSST and perhaps Struck, managed to surpass the best RBTs even for larger t_p . The superiority of DSST and Struck over other OLTs is consistent with results on VOT dataset [36]. However, the very poor performance of GOTURN [28], which is one of the best trackers on that dataset, indicates a fundamental difference in the challenges involved in the two paradigms of tracking. The speed plot shows another reason why OLTs are not suitable for high speed tracking applications - they are 10 to 30 times slower than RBTs except PF and RANSAC that are not implemented efficiently yet. It is not surprising that tracking based SLAM systems like SVO [22] use registration based trackers as they need to track hundreds of patches per frame.

To conclude the analysis in this section, we tested the performance of different SSMs against each other and the results are reported in the bottom right subplot of Fig. 1 using ESM with SSIM. Contrary to expectations, lower DOF SSMs are not better except perhaps affine and similitude that have slightly higher SR than homography for larger t_p . In fact, the SR curves of all low DOF SSMs approach those of homography as t_p increases which does indicate their higher robustness - though not as precise as homography, they tend to be more resistant to complete failure. Also, all three parameterizations of homography have almost identical performance, with their plots showing near perfect overlap. This suggests that the theoretical justification [9] for using ESM only with $SL(3)$ has little practical significance.

4.4.3 Search Methods

SR plots comparing different SMs for each AM are presented in the supplementary to save space as they contain the same data as Fig. 1. First fact to observe is that the four variants of LK do not perform identically. Though

FCLK and FALK are indeed evenly matched, both ICLK and IALK are significantly worse, with ICLK being notably better than IALK. This is especially true for CCRE where it outperforms both additive SMs. This contradicts the equivalence between these variants that was reported in [4] and justified there using both theoretical analysis and experimental results. The latter, however, were only performed on synthetic images and even the former used several approximations. Therefore, it is perhaps not surprising that this equivalence does not hold under real world conditions.

Secondly, ESM fails to outperform FCLK for any AM except SCV and ZNCC and even there it does not lead by much. This fact too emerges in contradiction to the theoretical analysis in [9] where ESM was shown to have second order convergence and so should be better than first order methods like FCLK. Thirdly, both additive LK variants, and especially IALK, fare worse against compositional ones when using robust AMs compared to SSD-like AMs. This is probably because the Hessian after convergence approach used for extending Newton’s method to these AMs does not make as much sense for additive formulations [16]. Lastly, PFFC is the best performing SM followed by NNIC and RKLT which proves the efficacy of composite SMs.

5. Conclusions

We presented two new similarity measures for registration based tracking and also formulated a novel way to decompose trackers in this domain into three sub modules. We tested many different combinations of methods for each sub module to gain interesting insights into their strengths and weaknesses. Several surprising results were obtained that proved previously published theoretical analysis to be somewhat inaccurate in practice, thus demonstrating the usefulness of this approach. Finally, the open source tracking framework used for generating these results was made publicly available so these can be easily reproduced.

References

- [1] Eigen: A C++ template library for linear algebra. <http://eigen.tuxfamily.org>. Accessed: 2017-01-10.
- [2] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 798–805. IEEE, 2006.
- [3] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *CVPR, IEEE Conference on*, volume 1, pages I–1090–I–1097 vol.1, 2001.
- [4] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, 56(3):221–255, Feb 2004.
- [5] A. Bartoli. Direct image registration with gain and bias. *Contributions au recalage d'images et à la reconstruction 3D de scènes rigides et déformables*, page 4, 2006.
- [6] A. Bartoli. Groupwise geometric and photometric direct image registration. *PAMI, IEEE Transactions on*, 30(12):2098–2108, 2008.
- [7] S. Benhimane, A. Ladikos, V. Lepetit, and N. Navab. Linear and quadratic subsets for template-based tracking. In *CVPR*, pages 1–6. IEEE, 2007.
- [8] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 943–948 vol.1, Sept 2004.
- [9] S. Benhimane and E. Malis. Homography-based 2D Visual Tracking and Servoing. *Int. J. Rob. Res.*, 26(7):661–676, July 2007.
- [10] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI, IEEE Transactions on*, (6):567–585, 1989.
- [11] R. Brooks and T. Arbel. Generalizing Inverse Compositional and ESM Image Alignment. *IJCV*, 87(3):191–212, May 2010.
- [12] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.
- [13] L. Čehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? In *IEEE Winter Conference on Applications of Computer Vision*, pages 540–547. IEEE, 2014.
- [14] D. Cobzas, M. Jagersand, and P. Sturm. 3D SSD tracking with estimated 3D planes. *Image and Vision Computing*, 27(1):69–79, 2009.
- [15] A. Crivellaro and V. Lepetit. Robust 3d tracking with descriptor fields. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3414–3421. IEEE, 2014.
- [16] A. Dame. *A unified direct approach for visual servoing and visual tracking using mutual information*. PhD thesis, University of Rennes, 2010.
- [17] A. Dame and E. Marchand. Accurate real-time tracking using mutual information. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 47–56, 2010.
- [18] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.
- [19] T. Dick, C. Perez, M. Jagersand, and A. Shademan. Real-time Registration-Based Tracking via Approximate Nearest Neighbour Search. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.
- [20] N. Dowson and R. Bowden. Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation. *PAMI*, 30(1):180–185, Jan 2008.
- [21] H. Firouzi and H. Najjaran. Adaptive on-line similarity measure for direct visual tracking. *Image and Vision Computing*, 32(4):227 – 236, 2014.
- [22] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.
- [23] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International journal of computer vision*, 94(3):335–360, 2011.
- [24] S. Gu, Y. Zheng, and C. Tomasi. Efficient visual object tracking with online nearest neighbor classifier. In *Computer vision—ACCV 2010*, pages 271–282. Springer, 2010.
- [25] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.
- [26] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *2011 International Conference on Computer Vision*, pages 263–270. IEEE, 2011.
- [27] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, March 2004.
- [28] D. Held, S. Thrun, and S. Savarese. *Learning to Track at 100 FPS with Deep Regression Networks*, pages 749–765. Springer International Publishing, Cham, 2016.
- [29] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.
- [30] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.
- [31] E. Ito, T. Okatani, and K. Deguchi. Accurate and robust planar tracking based on a model of image sampling and reconstruction process. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 1–8. IEEE, 2011.
- [32] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005.
- [33] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.
- [34] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, #jul# 2012.

- [35] Y. Keller and A. Averbuch. Global parametric image alignment via high-order approximation. *Computer Vision and Image Understanding*, 109(3):244–259, 2008.
- [36] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, et al. *The Visual Object Tracking VOT2016 Challenge Results*, pages 777–823. Springer International Publishing, Cham, 2016.
- [37] J. Kwon, H. S. Lee, F. C. Park, and K. M. Lee. A geometric particle filter for template-based visual tracking. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, 36(4):625–643, 2014.
- [38] J. Kwon, K. M. Lee, and F. C. Park. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 991–998. IEEE, 2009.
- [39] M. Li, T. Tan, W. Chen, and K. Huang. Efficient object tracking by incremental self-tuning particle filtering on the affine group. *IEEE Transactions on Image Processing*, 21(3):1298–1313, 2012.
- [40] Q. Liu, X. Zhao, and Z. Hou. Survey of single-target visual tracking methods based on online learning. *IET Computer Vision*, 8(5):419–428, October 2014.
- [41] A. Łoza, L. Mihaylova, D. Bull, and N. Canagarajah. Structural similarity-based object tracking in multimodality surveillance videos. *Machine Vision and Applications*, 20(2):71–83, 2009.
- [42] A. Loza, F. Wang, M. A. Patricio, J. García, and J. M. Molina. Optimised Particle Filter Approaches to Object Tracking in Video Sequences. In *Methods and Models in Artificial and Natural Computation*, pages 486–495. Springer, 2009.
- [43] A. Loza, F. Wang, J. Yang, and L. Mihaylova. Video object tracking with differential Structural SIMilarity index. In *ICASSP, IEEE International Conference on*, pages 1405–1408, May 2011.
- [44] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *7th International Joint Conference on Artificial intelligence*, volume 2, pages 674–679, 1981.
- [45] G. Nebehay and R. Pflugfelder. Clustering of Static-Adaptive correspondences for deformable object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2784–2791, 2015.
- [46] Y. Park, V. Lepetit, and W. Woo. ESM-Blur: Handling & rendering blur in 3D tracking and augmentation. In *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 163–166. IEEE Computer Society, 2009.
- [47] R. Richa, M. Souza, G. Scandaroli, E. Comunello, and A. von Wangenheim. Direct visual tracking under extreme illumination variations using the sum of conditional variance. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 373–377, Oct 2014.
- [48] R. Richa, R. Sznitman, R. Taylor, and G. Hager. Visual tracking using the sum of conditional variance. In *IROS, IEEE/RSJ International Conference on*, pages 2953–2958, Sept 2011.
- [49] G. H. Rogerio Richa, Raphael Sznitman. Robust Similarity Measures for Gradient-based Direct Visual Tracking. Technical report, CIRL, June 2012.
- [50] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *IJCV*, 77(1-3):125–141, May 2008.
- [51] A. Roy, X. Zhang, N. Wolleb, C. Perez Quintero, and M. Jagersand. Tracking benchmark and evaluation for manipulation tasks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2448–2453. IEEE, 2015.
- [52] L. Ruthotto. Mass-preserving registration of medical images. *German Diploma Thesis (Mathematics), Institute for Computational and Applied Mathematics, University of Münster*, 2010.
- [53] G. G. Scandaroli, M. Meilland, and R. Richa. Improving NCC-based Direct Visual Tracking. In *ECCV*, pages 442–455. Springer, 2012.
- [54] A. Shashua and Y. Wexler. Q-warping: Direct computation of quadratic reference surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):920–925, 2001.
- [55] H.-Y. Shum and R. Szeliski. Construction of Panoramic Image Mosaics with Global and Local Alignment. *IJCV*, 36(2):101–130.
- [56] G. Silveira. Photogeometric Direct Visual Tracking for Central Omnidirectional Cameras. *Journal of Mathematical Imaging and Vision*, 48(72), October 2014.
- [57] G. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In *CVPR. IEEE Conference on*, pages 1–6, 2007.
- [58] G. Silveira and E. Malis. Visual servoing from robust direct color image registration. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5450–5455. IEEE, 2009.
- [59] G. Silveira and E. Malis. Unified Direct Visual Tracking of Rigid and Deformable Surfaces Under Generic Illumination Changes in Grayscale and Color Images. *International Journal of Computer Vision*, 89(1):84–105, 2010.
- [60] G. Silveira and E. Malis. Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. *International journal of computer vision*, 89(1):84–105, 2010.
- [61] A. Singh and M. Jagersand. Modular Tracking Framework: A Unified Approach to Registration based Tracking. [arXiv:1602.09130 \[cs.CV\]](https://arxiv.org/abs/1602.09130), 2016.
- [62] A. Singh, A. Roy, X. Zhang, and M. Jagersand. Modular Decomposition and Analysis of Registration based Trackers. In *CRV*, June 2016.
- [63] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468, 2014.
- [64] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.
- [65] R. Szeliski and J. Coughlan. Spline-based image registration. *IJCV*, 22(3):199–218, 1997.

- [66] M. Trummer, J. Denzler, and C. Munkelt. Guided KLT Tracking Using Camera Parameters in Consideration of Uncertainty. In *International Conference on Computer Vision and Computer Graphics*, pages 252–261. Springer, 2008.
- [67] F. Wang and B. C. Vemuri. Non-rigid multi-modal image registration using cross-cumulative residual entropy. *IJCV*, 74(2):201–215, 2007.
- [68] X. Wang, C. Ning, A. Shi, and G. Lv. An improved similarity measure in particle filters for robust object tracking. In *Image and Signal Processing (CISP), 2013 6th International Congress on*, volume 1, pages 46–50. IEEE, 2013.
- [69] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, April 2004.
- [70] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *European Conference on Computer Vision*, pages 864–877. Springer, 2012.
- [71] X. Zhang, A. Singh, and M. Jagersand. RKLT: 8 DOF real-time robust video tracking combining coarse RANSAC features and accurate fast template registration. In *CRV*, pages 70–77. IEEE, 2015.
- [72] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):677–692, 2009.